

An Abstract Factorization Theorem for Explicit Substitutions

Beniamino Accattoli

INRIA & LIX (École Polytechnique), France
beniamino.accattoli@inria.fr

Abstract

We study a simple form of standardization, here called factorization, for explicit substitutions calculi, i.e. lambda-calculi where beta-reduction is decomposed in various rules. These calculi, despite being non-terminating and non-orthogonal, have a key feature: each rule terminates when considered separately. It is well-known that the study of rewriting properties simplifies in presence of termination (e.g. confluence reduces to local confluence). This remark is exploited to develop an abstract theorem deducing factorization from some axioms on local diagrams. The axioms are simple and easy to check, in particular they do not mention residuals. The abstract theorem is then applied to some explicit substitution calculi related to Proof-Nets. We show how to recover standardization by levels, we model both call-by-name and call-by-value calculi and we characterize linear head reduction via a factorization theorem for a linear calculus of substitutions.

1998 ACM Subject Classification F.4.1 Mathematical Logic – Lambda calculus and related systems, F.4.2 Grammars and Other Rewriting Systems

Keywords and phrases λ -calculus, Standardization, Explicit Substitutions, Abstract rewriting, Diagrammatic reasoning

Digital Object Identifier 10.4230/LIPIcs.RTA.2012.6

Category Regular Research Paper

1 Introduction

Background. The study of the rewriting properties of a system S simplifies considerably if there cannot be infinite reductions, i.e. if the rewriting relation \rightarrow_S associated to S is strongly normalizing. The typical example is the study of confluence, which can be reduced to local confluence in presence of strong normalization, thanks to Newman's lemma. Unfortunately, many interesting systems— λ -calculus for instance—do not enjoy strong normalization (\mathcal{SN})¹.

There is a special class of non strongly normalizing rewriting systems which is given by those systems having more than one rewriting rule and s.t. any of their rules is \mathcal{SN} when considered alone; they could be called *locally terminating systems*. Natural examples are most λ -calculi where β -reduction is decomposed into a set of more atomic rules. This is typically done by extending the syntax of λ -calculus with a new term constructor $t[x/u]$ (sometimes written **let** $x = u$ **in** t) denoting a delayed or **explicit substitution** (ES), and decomposing the β -rule:

$$(\lambda x.t) u \rightarrow_{\beta} t\{x/u\}$$

¹ λ -calculus is an orthogonal system, and so confluence of λ -calculus is easy for other reasons. But the systems we will deal with in this paper are not orthogonal.



© Beniamino Accattoli;

licensed under Creative Commons License NC-ND

23rd International Conference on Rewriting Techniques and Applications (RTA'12).

Editor: A. Tiwari; pp. 6–21



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



into two rules:

$$(\lambda x.t) u \rightarrow_B t[x/u] \rightarrow_S t\{x/u\}$$

In the ES-literature \rightarrow_S is in turn decomposed into a set of rules (see [22] for a survey), but for the present discussion this is not relevant. Explicit substitutions were introduced to close the gap between the theory of λ -calculus and implementations [1]. Their rewriting theory has also been the subject of a lot of research, after Melliès showed the possibility of pathological behaviors [25].

A similar decomposition of \rightarrow_β arise in many contexts, not necessarily in relation to implementations and often without explicit reference to ES. Indeed, it can be found in studies on call-by-value [19], complexity classes [32], linear systems [21], side-effects [20], etc. Usually, these refined systems are not orthogonal nor strongly normalizing, and their rewriting theory may be involved.

Local to global, via \mathcal{SN} . In locally terminating systems termination of each rule can sometimes be used to reduce *global* rewriting properties to a *local* form, similarly to how Newman's lemma reduces confluence to local confluence. The aim of the paper is to reduce a simple form of standardization for ES-calculi to a local form, using as key ingredient termination of each single rule.

Termination and confluence concern the *existence* and the *uniqueness* of normal forms, which are the results of a computation. Standardization instead is about *how to compute*: it identifies a specific class of reductions to which any other reduction can be transformed by permuting its steps. It has many important corollaries, in particular it gives a normalizing strategy for evaluation (see [9]).

Factorization in λ -calculus. It is well-known that in λ -calculus any reduction $t \rightarrow_\beta^* u$ can be re-organised so that

$$t \rightarrow_h^* \rightarrow_i^* u$$

i.e. by first reducing on the head (\rightarrow_h^*) and then everywhere else (\rightarrow_i^*). Concisely this is expressed by $\rightarrow_\beta^* = (\rightarrow_h \cup \rightarrow_i)^* \subseteq \rightarrow_h^* \rightarrow_i^*$. This is not exactly the standardization theorem: it is an easier *factorization* theorem which can be inferred from standardization [27] or it can be used to prove it [30, 9]. "Conceptually, the factorisation property means that the efficient part of a computation can always be separated from its junk" [27]. Applications of standardization often only require factorization, *e.g.* the operational characterization of solvability for λ -calculus ([9]).

At first sight factorization is easy: a two-steps sequence as $\rightarrow_i \rightarrow_h$ can always be permuted, and the theorem seems to follow from this local permutation property. The swap is indeed possible, but it is non-linear, *i.e.* it has the following general form $\rightarrow_i \rightarrow_h \subseteq \rightarrow_h^+ \rightarrow_i^*$, which can diagrammatically be represented as:

$$\begin{array}{ccccc} t & \xrightarrow{\quad} & i & & u \\ \downarrow & \swarrow & & & \downarrow \\ v & \xrightarrow{\quad} & i^* & & w \end{array}$$

Now, there are two easy abstract lemmas (Lemma 4.2 and 4.3), similar to Newman's lemma, which would imply the factorization theorem *but only when* either the sequence \rightarrow_h^+ is composed of at most one step or \rightarrow_h is strongly normalizing. Unfortunately, neither of these conditions holds for λ -calculus. Indeed, consider the following diagram (where $I = \lambda y.y$):

$$\begin{array}{ccccc} (\lambda x.x x) (I I) & \xrightarrow{\quad} & i & & (\lambda x.x x) I \\ \downarrow & \swarrow & & & \downarrow \\ (I I) (I I) & \xrightarrow{\quad} & h & & I I \end{array}$$

It shows that in general \rightarrow_h^+ can have length greater than one². And there exist \rightarrow_h -divergent terms, for instance $(\lambda x.(x x)) \lambda x.(x x)$. Therefore, factorization in non-terminating systems is non-trivial.

The rewriting technique. Let us sketch the idea from which this paper stems. The decomposition of β into B and s induces head (noted $\rightarrow_{B\circ}$ and $\rightarrow_{s\circ}$) and internal ($\rightarrow_{B\bullet}$ and $\rightarrow_{s\bullet}$) variants of B and s . The new head reduction \rightarrow_\circ , given by $\rightarrow_\circ := \rightarrow_{B\circ} \cup \rightarrow_{s\circ}$, is not strongly normalizing, but now it splits into two strongly normalizing reductions, $\rightarrow_{B\circ}$ and $\rightarrow_{s\circ}$. The internal-head permutation splits into four diagrams (the dotted lines are not specified on purpose):

$$\begin{array}{cccc}
 \begin{array}{c} t \\ \vdots \\ \downarrow_* \\ v \end{array} & \xrightarrow{B\bullet} & \begin{array}{c} u \\ \vdots \\ \downarrow_{B\circ} \\ w \end{array} & \begin{array}{c} t \\ \vdots \\ \downarrow_* \\ v \end{array} \xrightarrow{s\bullet} \begin{array}{c} u \\ \vdots \\ \downarrow_{s\circ} \\ w \end{array} \\
 \swarrow & & \searrow & \swarrow & \searrow \\
 v & \dashrightarrow^* & w & v & \dashrightarrow^* & w
 \end{array}$$

These diagrams can now be studied one by one using strong normalization of $\rightarrow_{B\circ}$ and $\rightarrow_{s\circ}$, which allows to apply the abstract lemmas (Lemma 4.2 and 4.3, and actually a new abstract lemma, Lemma 4.4, is also needed). The obtained factorizations can then easily be glued together, getting the factorization theorem for \rightarrow_\circ . Therefore factorization—surprisingly—turns out to be easier in ES-calculi than in λ -calculus. Actually, in λ -calculus it is possible to prove factorization using some termination assumptions, by considering complete developments of given sets of redexes, which are always finite (see [9], Chapter 11.4). However, this technique requires to define complex notions such as residuals and developments, while our approach circumvents their need.

The main result of the paper is a set of axioms on local diagrams which imply the factorization theorem. Despite being conceived for ES-calculi the axiomatization does not rely on any particular feature of such calculi, it is simply formulated in terms of swaps between reduction relations. Moreover, the axioms are simple and can be checked without using heavy techniques as positions, labels or underlinings. Last, the theorem follows from the axioms quite easily, once the right abstract lemmas for postponement have been isolated.

Applications. We developed the abstract result of this paper while studying standardization for some λ -calculi related to Linear Logic Proof-Nets. Starting from these graphical syntaxes, a new *at a distance* approach to explicit substitutions has recently been proposed [5]. Distance-calculi are simpler than those of the earlier generation, in particular they have less rules and they led to new results on λ -calculus [5, 6]. We present various concrete applications of our abstract theorem to three such ES-calculi:

1. **The substitution calculus λ_{sub} :** the head reduction of λ_{sub} corresponds to reduction at level 0 in Proof-Nets. We prove factorization and show how to iterate it in order to get a standardization theorem by levels in the style of Linear Logic Proof-Nets.
2. **The value substitution calculus λ_{vsub} ,** a call-by-value version of λ_{sub} . We factorize λ_{vsub} reduction with respect to weak reduction, which plays the same role as head reduction in the call-by-name case. We also factorize λ_{vsub} reduction with respect to another scheme, obtained by mixing head and weak reduction. These factorizations have been used in [8] to give a new characterization of call-by-value solvability.
3. **The linear substitution λ -calculus λ_{lsub} :** it is a natural refinement of λ_{sub} where explicit substitutions act on one occurrence at the time. It is also a slight variation over

² It is enough to replace $\lambda x.x x$ with $\lambda x.(x x) x$ in the example to get that also \rightarrow_i^+ has length greater than one

a calculus proposed by Robin Milner (and related to Bigraphs and Pure Proof-Nets), and a simplification of the structural calculus λ_j of [5]. Surprisingly, the reductions by levels of λ_{sub} do not induce factorization nor standardization for λ_{sub} . That approach is then refined: the proper head reduction of λ_{sub} is head *linear* reduction, an important notion having strong connections with Krivine's abstract machine (KAM) [14], Proof-Nets [23], Geometry of Interaction [13], Game Semantics [12], Differential λ -calculus [16] and the translation of λ -calculus into the π -calculus [24]. Ours is the first characterization of linear head reduction from a rewriting point of view. Moreover, we generalize linear head reduction into an indexed family of linear reductions, and give a generalized factorization theorem for λ_{sub} .

Related work. Standardization was first studied axiomatically by Gonthier, Levy and Mellies in [18], and then further explored by Mellies [26, 28], who also gives axioms for factorization in [27]. Our approach differs in two main points: 1) his theory is very general and powerful, but also complex and difficult to apply, while ours, having been designed around a specific class of systems, is easy to apply, but of course its scope is limited; 2) we use factorization to prove standardization while he proceeds the other way around.

Road-map. In Section 2 we fix some terminology and notation. In Section 3 we define the substitution calculus λ_{sub} and use it to introduce our axiomatics. Section 4 recalls and develops some abstract tools about postponement of reductions. In Section 5 any system satisfying the axioms is shown to enjoy a factorization theorem. In Section 6 we prove two factorizations for the value-substitution calculus λ_{vsub} . Section 7 introduces the linear substitution calculus λ_{lsub} , factorization with respect to linear head reduction, and then generalizes linear head reduction. Section 8 concludes.

2 Notations and conventions

We follow a relational approach to abstract rewriting, *i.e.* we see a rewriting rule \rightarrow on a set S as a subset of $\rightarrow \subseteq S \times S$. Composition of rewriting steps is simple juxtaposition, therefore $\rightarrow_1 \rightarrow_2$ stands for a sequence like $t \rightarrow_1 u \rightarrow_2 v$. In the following we use $\rightarrow_1 \rightarrow_2 \subseteq \rightarrow_3 \rightarrow_4$ as a compact notation for a diagram like:

$$\begin{array}{ccccc} t & & \rightarrow_1 & & u \\ \downarrow & & \swarrow & & \downarrow \\ \downarrow_3 & & & & \downarrow_2 \\ v & & \dashrightarrow_4 & & w \end{array}$$

Composition of rewriting rules has precedence over union, *i.e.* $\rightarrow_1 \rightarrow_2 \cup \rightarrow_3$ should be read as $(\rightarrow_1 \rightarrow_2) \cup \rightarrow_3$. Given a rewriting relation \rightarrow we note \rightarrow^+ and \rightarrow^* the transitive and transitive-reflexive closure of \rightarrow , respectively. We will note \rightarrow_\circ and \rightarrow_\bullet the head and internal variant of a given relation \rightarrow . Unfortunately, the paper requires a huge number of rewriting relations.

3 The substitution calculus λ_{sub}

The calculus λ_{sub} is given by the following syntax for terms:

$$t, u, s, v, r ::= x \mid \lambda x.t \mid t \ u \mid t[x/u]$$

The constructor $t[x/u]$ is called an **explicit substitution**. Both $\lambda x.t$ and $t[x/u]$ bind x in t . We use L to note a possibly empty list of explicit substitutions $[x_1/u_1] \dots [x_k/u_k]$. The

rewriting relation $\rightarrow_{\lambda_{\text{sub}}}$ of λ_{sub} is given by the union of rules \rightarrow_{dB} and \rightarrow_{s} , which are obtained as the context closure of following two root rules:

$$(\lambda x.t)L s \mapsto_{\text{dB}} t[x/s]L \qquad t[x/u] \mapsto_{\text{s}} t\{x/u\}$$

Rule \rightarrow_{dB} is the rule labeled **B** at page 1 but here acting *at a distance* (which is the reason for **d**): the function $\lambda x.t$ and the argument s can interact even if there is L between them. This is motivated by the close relation between λ_{sub} and graphical formalisms as (Pure) Proof-Nets or λ j-dags, see [3, 4]. Morally, \rightarrow_{dB} (resp. \rightarrow_{s}) corresponds to the multiplicative (resp. exponential) reduction of Proof-Nets. With different motivations this rule was also considered by Curien, Hardin and Lévy in [11].

The substitution calculus may be suspected to be a useless modification of the λ -calculus, because it has explicit substitutions but it executes them in just one shot. However, in [6] λ_{sub} is the crucial tool in the proofs of some results on λ -calculus for which no other proof is known. It should be taken as a tool to study λ -calculus rather than implementations.

The reduction relation $\rightarrow_{\lambda_{\text{sub}}}$ of λ_{sub} is not \mathcal{SN} , since $\rightarrow_{\beta} \subseteq \rightarrow_{\lambda_{\text{sub}}} \rightarrow_{\lambda_{\text{sub}}}$. However, both \rightarrow_{dB} and \rightarrow_{s} are \mathcal{SN} .

► **Lemma 3.1** ([6]). *The reductions \rightarrow_{dB} and \rightarrow_{s} are strongly normalizing and confluent. Moreover, $\rightarrow_{\lambda_{\text{sub}}}$ is confluent.*

The literature on λ -calculus usually factors β reduction with respect to head and internal reductions. We are going to do the same in λ_{sub} , but our notion of head reduction will be slightly more liberal than the usual one. **Head contexts** H are defined by:

$$H ::= [\cdot] \mid \lambda x.H \mid H t \mid H[x/t]$$

The reductions $\rightarrow_{\text{dB} \circ}$ and $\rightarrow_{\text{s} \circ}$ are defined as the closure by *head* contexts of \mapsto_{dB} and \mapsto_{s} . Then λ_{sub} head reduction is noted $\rightarrow_{\lambda_{\text{sub}} \circ}$ and given by $\rightarrow_{\text{dB} \circ} \cup \rightarrow_{\text{s} \circ}$.

Our head reduction is a non-deterministic strategy. Consider:

$$(\lambda x.y[y/I]) u \xrightarrow{\text{dB}} (\lambda x.(I I)) u \xrightarrow{\text{dB}} (I I)[x/u]$$

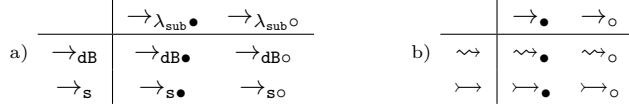
However, a simple case analysis shows that $\rightarrow_{\lambda_{\text{sub}} \circ}$ enjoys the diamond property, which is just a more abstract way to say that its non-determinism is harmless: it does not affect reduction lengths and $\rightarrow_{\lambda_{\text{sub}} \circ}$ normalizes weakly iff it normalizes strongly; in fact $\rightarrow_{\lambda_{\text{sub}} \circ}$ can be considered as a deterministic strategy. In particular, reducing $\rightarrow_{\lambda_{\text{sub}} \circ}$ in an outermost way recovers a more traditional formulation of head reduction. A motivation for our approach is that $\rightarrow_{\lambda_{\text{sub}} \circ}$ corresponds exactly to reduction at level 0 in Linear Logic Proof-Nets.

In order to study factorization with respect to head reduction we need to define the complements of $\rightarrow_{\lambda_{\text{sub}} \circ}$, $\rightarrow_{\text{dB} \circ}$ and $\rightarrow_{\text{s} \circ}$. We define the **internal reduction** $\rightarrow_{\lambda_{\text{sub}} \bullet}$ as the union of $\rightarrow_{\text{dB} \bullet}$ and $\rightarrow_{\text{s} \bullet}$, where $\rightarrow_{\text{dB} \bullet} := \rightarrow_{\text{dB}} \setminus \rightarrow_{\text{dB} \circ}$ and $\rightarrow_{\text{s} \bullet} := \rightarrow_{\text{s}} \setminus \rightarrow_{\text{s} \circ}$. It is possible to show that $\rightarrow_{\text{dB} \bullet}$ and $\rightarrow_{\text{s} \bullet}$ can be defined as the closure by internal contexts of \mapsto_{dB} and \mapsto_{s} , where **internal contexts** I are defined by:

$$I ::= t C \mid t[x/C] \mid H[I]$$

where C is no matter which context (and the notation $H[I]$ denotes the context obtained substituting the hole of H with I). The reductions are summarized by the square in Fig. 1.a, where each coordinate is equal to the union of the row/column it denotes.

The local internal-head permutations are given by the following lemma, proved by induction on the internal reductions and by means of a substitution lemma (details in [2]):



■ **Figure 1** Squares of reductions

► **Lemma 3.2** (head-internal diagrams). *The following swaps hold:*

1. $\rightarrow_{\text{dB}\bullet} \rightarrow_{\text{dB}\circ} \subseteq \rightarrow_{\text{dB}\circ} \rightarrow_{\text{dB}\bullet}$.
2. $\rightarrow_{\text{s}\bullet} \rightarrow_{\text{s}\circ} \subseteq \rightarrow_{\text{s}\circ} \rightarrow_{\text{s}\bullet}^* \cup \rightarrow_{\text{s}\circ} \rightarrow_{\text{s}\bullet}^*$.
3. $\rightarrow_{\text{s}\bullet} \rightarrow_{\text{dB}\circ} \subseteq \rightarrow_{\text{dB}\circ} \rightarrow_{\text{s}\bullet}$.
4. $\rightarrow_{\text{dB}\bullet} \rightarrow_{\text{s}\circ} \subseteq \rightarrow_{\text{s}\circ} \rightarrow_{\text{dB}\circ} \rightarrow_{\text{dB}\bullet}^* \cup \rightarrow_{\text{s}\circ} \rightarrow_{\text{dB}\bullet}^*$.

Points 1-3 give swaps for which we can apply some abstract lemmas in the literature, since $\rightarrow_{\text{dB}\circ}$ and $\rightarrow_{\text{s}\circ}$ are strongly normalizing, and easily get factorization. However, Point 4 introduces a new behaviour: swapping $\rightarrow_{\text{dB}\bullet} \rightarrow_{\text{s}\circ}$ can yield $\rightarrow_{\text{s}\circ} \rightarrow_{\text{dB}\circ} \rightarrow_{\text{dB}\bullet}^*$, *i.e.* a third kind of reduction ($\rightarrow_{\text{dB}\circ}$) can pop out. Let us give an example:

$$\begin{array}{ccc} (x\ x)[x/(\lambda z.z)\ y] & \xrightarrow{\text{dB}\bullet} & (x\ x)[x/z[z/y]] \\ \downarrow_{\text{s}\circ} & \searrow & \downarrow_{\text{s}\circ} \\ ((\lambda z.z)\ y)\ ((\lambda z.z)\ y) & \xrightarrow{\text{dB}\circ} & z[z/y]\ ((\lambda z.z)\ y) \xrightarrow{\text{dB}\bullet} z[z/y]\ z[z/y] \end{array}$$

Note that the diagram can be written as $\rightarrow_{\text{dB}\bullet} \rightarrow_{\text{s}\circ} \subseteq \rightarrow_{\text{s}\circ} \rightarrow_{\text{dB}\circ}^*$. This phenomenon cannot happen in λ -calculus, where there is only one head and one internal rewriting relation. Other explicit substitution calculi (e.g. those in Section 6 and 7) present similar (but not identical) forms of permutations. They can be grouped together under the notion of *square factorization system*.

Let us abstract \rightarrow_{dB} and \rightarrow_{s} as two reductions \rightsquigarrow and \rightarrow . Now, the square of reductions becomes as in Fig. 1.b, where we use the notations $\rightarrow_{\circ} := \rightsquigarrow_{\circ} \cup \rightarrow_{\circ}$, $\rightarrow_{\bullet} := \rightsquigarrow_{\bullet} \cup \rightarrow_{\bullet}$. Let us also set $\rightarrow := \rightarrow_{\circ} \cup \rightarrow_{\bullet} = \rightsquigarrow \cup \rightarrow$.

► **Definition 3.3.** A **square factorization system** (SFS) S is given by a set $|S|$ and four reduction relations ($\rightsquigarrow_{\bullet}, \rightsquigarrow_{\circ}, \rightarrow_{\bullet}, \rightarrow_{\circ}$) on $|S|$ s. t. the following conditions hold:

1. **Termination:** \rightsquigarrow_{\circ} and \rightarrow_{\circ} are strongly normalizing.
2. **Row-swap 1:** $\rightsquigarrow_{\bullet} \rightsquigarrow_{\circ} \subseteq \rightsquigarrow_{\circ}^+ \rightsquigarrow_{\bullet}^*$.
3. **Row-swap 2:** $\rightarrow_{\bullet} \rightarrow_{\circ} \subseteq \rightarrow_{\circ}^+ \rightarrow_{\bullet}^*$.
4. **Diagonal-swap 1:** $\rightarrow_{\bullet} \rightsquigarrow_{\circ} \subseteq \rightsquigarrow_{\circ} \rightarrow_{\bullet}^*$.
5. **Diagonal-swap 2:** $\rightsquigarrow_{\bullet} \rightarrow_{\circ} \subseteq \rightarrow_{\circ} \rightsquigarrow_{\bullet}^*$.

The names of axioms 2-5 refer to the square table in Fig. 1.b.

Note the presence of $\rightarrow = \rightarrow_{\circ} \cup \rightarrow_{\bullet}$ in axiom 4 and $\rightsquigarrow = \rightsquigarrow_{\circ} \cup \rightsquigarrow_{\bullet}$ in axiom 5: the diagonal swaps are where a third reduction pops out. Interestingly, the axioms are symmetric, *i.e.* swapping the roles of \rightarrow and \rightsquigarrow one still has a SFS.

In Section 5 we prove that any SFS enjoys a factorization theorem, *i.e.* $\rightarrow^* \subseteq \rightarrow_{\circ}^* \rightarrow_{\bullet}^*$ holds for any SFS. The important assumption in order to handle the diagonal swaps is that the anticipated relation (e.g. \rightsquigarrow_{\circ} in axiom 4) occurs only once in the permuted reduction. Now, lemmas 3.2 and 3.1 prove that $(\rightarrow_{\text{dB}\bullet}, \rightarrow_{\text{dB}\circ}, \rightarrow_{\text{s}\bullet}, \rightarrow_{\text{s}\circ})$ is a SFS, therefore by the abstract factorization result in Section 5 (Theorem 5.2.2, page 15) it follows:

► **Corollary 3.4.** $\rightarrow_{\lambda_{\text{sub}}}^* \subseteq \rightarrow_{\lambda_{\text{sub}}\circ}^* \rightarrow_{\lambda_{\text{sub}}\bullet}^*$.

It is natural to wonder if from factorization for λ_{sub} one can recover the usual factorization for λ -calculus. Let us sketch an answer. To every λ_{sub} term t one can associate the λ -term $\mathbf{s}(t)$, which is the normal form with respect to $\rightarrow_{\mathbf{s}}$ and it is easy to show that if $t \rightarrow_{\lambda_{\text{sub}}} t'$ then $\mathbf{s}(t) \rightarrow_{\beta}^* \mathbf{s}(t')$. Such a projection does not map factorized reductions in λ_{sub} to factorized reduction in λ -calculus, because internal steps may project on head steps. Consider for instance the reduction $t = x[x/I \ I] \rightarrow_{\text{dB}} x[x/y[y/I]] = u$ whose projection is $\mathbf{s}(t) = I \ I \rightarrow_{\mathbf{h}} I = \mathbf{s}(u)$. However, this mismatch arises only if reductions take place inside explicit substitutions. Now, a reduction ρ in λ -calculus induces a reduction ρ' in λ_{sub} where every β -step is simulated by a \rightarrow_{dB} step followed by a $\rightarrow_{\mathbf{s}}$ step, and in ρ' reductions never take place inside substitutions. It is possible to show that the factorization of ρ' gives a reduction τ with the same properties of ρ' , whose projection is then a factorized reduction in λ -calculus. But the proof of this fact is left to a longer version of this paper.

We now show how to generalize the factorization theorem for λ_{sub} into a standardization theorem. In terms of Proof-Nets the factorization theorem says that it is always possible to first reduce at level 0 and then inside arguments of applications and substitutions, which in Proof-Nets are wrapped into !-boxes. The head factorization can be iterated, getting standardization *by levels* in the style of Linear Logic Proof-Nets.

► **Definition 3.5** (Boxed contexts). Boxed contexts are indexed by an integer $i \in \mathbb{N}$ called the **level** of the boxed context and defined by:

$$B_0 ::= H \qquad B_{i+1} ::= t \ B_i \mid t[x/B_i] \mid B_0[B_{i+1}]$$

The closure with respect to B_0 —which may be confusing—is used to assign level 1 to contexts as $(\lambda x.t \ [\cdot])[x/u]$, for instance. Now, define $\xrightarrow{\mathbf{s}}^i := B_i[\mapsto_{\mathbf{s}}]$, $\xrightarrow{\text{dB}}^i := B_i[\mapsto_{\text{dB}}]$ and $\xrightarrow{\lambda_{\text{sub}}}^i := \xrightarrow{\mathbf{s}}^i \cup \xrightarrow{\text{dB}}^i$. Moreover, define $\xrightarrow{\mathbf{s}}^i := \bigcup_{j>i} \xrightarrow{\mathbf{s}}^j$, and similarly for $\xrightarrow{\text{dB}}^i$ and $\xrightarrow{\lambda_{\text{sub}}}^i$ (and for \geq). According to our definitions we have that $\rightarrow_{\lambda_{\text{sub}} \circ} = \xrightarrow{\lambda_{\text{sub}}}^0$ and $\rightarrow_{\lambda_{\text{sub}} \bullet} = \xrightarrow{\lambda_{\text{sub}}}^0$. Lemma 3.2 can be generalized as follows:

► **Lemma 3.6** (Factorization by levels). *The following swaps hold:*

1. $\xrightarrow{\text{dB}}^i \xrightarrow{\text{dB}}^i \subseteq \xrightarrow{\text{dB}}^i \xrightarrow{\text{dB}}^i$.
2. $\xrightarrow{\mathbf{s}}^i \xrightarrow{\mathbf{s}}^i \subseteq \xrightarrow{\mathbf{s}}^i \xrightarrow{\mathbf{s}}^i$.
3. $\xrightarrow{\mathbf{s}}^i \xrightarrow{\text{dB}}^i \subseteq \xrightarrow{\text{dB}}^i \xrightarrow{\mathbf{s}}^i$.
4. $\xrightarrow{\text{dB}}^i \xrightarrow{\mathbf{s}}^i \subseteq \xrightarrow{\mathbf{s}}^i \xrightarrow{\text{dB}}^i$.

Thus $(\xrightarrow{\text{dB}}^i, \xrightarrow{\text{dB}}^i, \xrightarrow{\mathbf{s}}^i, \xrightarrow{\mathbf{s}}^i)$ is a SFS and $\xrightarrow{\lambda_{\text{sub}}}^i = \xrightarrow{\lambda_{\text{sub}}}^i \xrightarrow{\lambda_{\text{sub}}}^i$.

Proof. We prove the first point, the others are proved analogously. If $t \xrightarrow{\text{dB}}^j u \xrightarrow{\text{dB}}^i v$ with $j > i$ then there are two cases. 1) there exists a context B_i s.t. $t = B_i[w]$, $u = B_i[w']$ and $v = B_i[w'']$ with $w \xrightarrow{\text{dB}}^j w' \xrightarrow{\text{dB}}^0 w''$, which is equivalent to $w \rightarrow_{\text{dB} \bullet} w' \rightarrow_{\text{dB} \circ} w''$. By Lemma 3.2.1 we get $w \rightarrow_{\text{dB} \circ} w' \rightarrow_{\text{dB} \bullet} w''$, i.e. $w \xrightarrow{\text{dB}}^0 w' \xrightarrow{\text{dB}}^0 w''$, and so $t \xrightarrow{\text{dB}}^i v$. 2) There is no such B_i , then the two steps are independent and permute. ◀

As a corollary we get the standardization theorem by levels for Linear Logic Proof-Nets [15]. Define the level k of a term u as the maximum k s.t. $u = B_k[v]$ for some v .

► **Corollary 3.7** (standardization by levels). *If $t \rightarrow_{\lambda_{\text{sub}}}^* u$ and k is the level of u then $t \xrightarrow{\lambda_{\text{sub}}}^0 \xrightarrow{\lambda_{\text{sub}}}^1 \dots \xrightarrow{\lambda_{\text{sub}}}^{k-1} \xrightarrow{\lambda_{\text{sub}}}^k u$.*

Proof. By definition $\rightarrow_{\lambda_{\text{sub}}}^* = \geq_0^*$. By lemma 3.6 there exists u_0 s.t. $t \xrightarrow{0}_{\lambda_{\text{sub}}}^* u_0 \xrightarrow{0}_{\lambda_{\text{sub}}}^* u$. Then we iterate on $u_0 \xrightarrow{0}_{\lambda_{\text{sub}}}^* u$, getting that there exists u_1 s.t. $u_0 \xrightarrow{1}_{\lambda_{\text{sub}}}^* u_1 \xrightarrow{1}_{\lambda_{\text{sub}}}^* u$, and so on, until we get $t \xrightarrow{0}_{\lambda_{\text{sub}}}^* u_0 \xrightarrow{1}_{\lambda_{\text{sub}}}^* u_1 \dots \xrightarrow{k-1}_{\lambda_{\text{sub}}}^* u_{k-1} \xrightarrow{k}_{\lambda_{\text{sub}}}^* u_k \xrightarrow{k}_{\lambda_{\text{sub}}}^* u$. Now, consider $u_k \xrightarrow{k}_{\lambda_{\text{sub}}}^* u$. This reduction is empty, otherwise u would have level greater than k , which is absurd. \blacktriangleleft

The paper continues as follows: next section introduces some abstract lemmas for postponements of reductions; Section 5 proves the factorization theorem for SFSs in an abstract way. Sections 6 and 7 give some more applications of the factorization theorem.

4 An abstract toolbox for postponement

In this section we prove four abstract lemmas about permutation properties, which are used to prove the abstract factorization theorem in Section 5. Let S be a rewriting system having three reductions \rightarrow_1 , \rightarrow_2 , and \rightarrow_3 , and for $x, y \in \{1, 2, 3\}$ note $\rightarrow_{x,y} = \rightarrow_x \cup \rightarrow_y$.

Let us recall some rewriting notions from [31]. What we call factorization is a form of postponement: \rightarrow_2 **postpones** after \rightarrow_1 if $\rightarrow_{1,2}^* \subseteq \rightarrow_1^* \rightarrow_2^*$. Postponement is equivalent to commutation. Two reductions \rightarrow_1 and \rightarrow_2 **commute** iff:

$$\begin{array}{ccc} t & \xrightarrow{1}^* & u \\ \downarrow_{*2} & & \\ v & & \end{array} \quad \text{implies } \exists w \text{ s.t.} \quad \begin{array}{ccc} t & \xrightarrow{1}^* & u \\ \downarrow_{*2} & & \downarrow_{*2} \\ v & \xrightarrow{1}^* & w \end{array}$$

It is easy to show that \rightarrow_2 postpones after \rightarrow_1 iff \rightarrow_1 commutes with $\rightarrow_2 \leftarrow$. Therefore, lemmas about commutations can be turned into lemmas about postponement, and conversely. Commutation/postponement is a generalization of confluence, indeed a reduction \rightarrow is confluent iff it commutes with itself (iff \leftarrow postpones after \rightarrow).

The first lemma proves postponement from a semi-local hypothesis, *i.e.* from a swap of \rightarrow_2^* and \rightarrow_1 (and not of \rightarrow_2 and \rightarrow_1). Afterwards, we show two different local conditions for the semi-local hypothesis. This and the lemma which will follow are standard, we prove them explicitly only to help the unacquainted reader.

► **Lemma 4.1** (Semi-local condition for postponement). *If $\rightarrow_2^* \rightarrow_1 \subseteq \rightarrow_1^* \rightarrow_2^*$ then $\rightarrow_{1,2}^* \subseteq \rightarrow_1^* \rightarrow_2^*$.*

Proof. By induction on the number k of \rightarrow_1 steps in $\tau : t \rightarrow_{1,2}^* u$. The case $k = 0$ is trivial. Let $k > 0$. Then if τ is not of the form of the statement it has the following form: $t \rightarrow_1^* \rightarrow_2^* \rightarrow_1 \rightarrow_{1,2}^* u$. Using the hypothesis we get: $t \rightarrow_1^* \rightarrow_1^* \rightarrow_2^* \rightarrow_{1,2}^* u$. The *i.h.* on $\rightarrow_2^* \rightarrow_{1,2}^*$ gives: $t \rightarrow_1^* \rightarrow_1^* \rightarrow_1^* \rightarrow_2^* u$ and we conclude. \blacktriangleleft

The first local condition is that $\rightarrow_2 \rightarrow_1 \subseteq \rightarrow_1 \rightarrow_2^*$. Note that one step of \rightarrow_1 on the left becomes one step of \rightarrow_1 on the right. The right part can also be replaced by $\rightarrow_1 \rightarrow_2^* \cup \rightarrow_2^*$, but not by $\rightarrow_1^* \rightarrow_2^*$, otherwise the inductive argument does not work and it is not difficult to build a counter-example (see [9]).

► **Lemma 4.2** (Local condition for postponement). *If $\rightarrow_2 \rightarrow_1 \subseteq \rightarrow_1 \rightarrow_2^*$ then $\rightarrow_2^* \rightarrow_1 \subseteq \rightarrow_1 \rightarrow_2^*$ and so $\rightarrow_{1,2}^* \subseteq \rightarrow_1^* \rightarrow_2^*$.*

Proof. We prove the first consequence, the second one follows from the first and Lemma 4.1. By induction on k , where $t \rightarrow_2^k \rightarrow_1 u$. The case $k = 0$ is trivial. Let $k > 0$. Then $t \rightarrow_2^{k-1} \rightarrow_2 \rightarrow_1 u$ and applying the hypothesis to the suffix we get $t \rightarrow_2^{k-1} \rightarrow_1 \rightarrow_2^* u$. We conclude applying the *i.h.* to the prefix $\rightarrow_2^{k-1} \rightarrow_1$. \blacktriangleleft

In order to get a factorization theorem for SFSSs we have to deal with permutations of the form $\rightarrow_2 \rightarrow_1 \subseteq \rightarrow_1 \rightarrow_1 \rightarrow_2^*$, given for instance by Lemma 3.2.2. These commutations are harmless if \rightarrow_1 is strongly normalizing, as next lemma shows. The lemma has been independently proved by Geser in his Ph.D. thesis [17] and by Di Cosmo and Piperno in [10], where it is presented for commutation, and it is here adapted to postponement. It can be seen as a generalization of Newman's lemma.

► **Lemma 4.3** (Geser-Di Cosmo-Piperno's lemma, adapted). *If $\rightarrow_2 \rightarrow_1 \subseteq \rightarrow_1^+ \rightarrow_2^*$ and \rightarrow_1 is strongly normalizing then $\rightarrow_2^* \rightarrow_1^+ \subseteq \rightarrow_1^+ \rightarrow_2^*$, and so $\rightarrow_{1,2}^* \subseteq \rightarrow_1^* \rightarrow_2^*$.*

Proof. We prove the first consequence, the second one follows from the first and Lemma 4.1. Let $\tau : t \rightarrow_2^k \rightarrow_1^h u$. By induction³ on the pair $(\eta(t), k)$, where $\eta(t)$ is the length of the maximal \rightarrow_1 reduction from t , ordered lexicographically. The case $k = 0$ is trivial, then let $k > 0$. Now, if $\tau = \rightarrow_2^{k-1} \rightarrow_2 \rightarrow_1 \rightarrow_1^{h-1}$ then by applying the first hypothesis to the central subsequence we get $\tau \subseteq \rightarrow_2^{k-1} \rightarrow_1^+ \rightarrow_2^* \rightarrow_1^{h-1}$. The measure of the prefix $\rightarrow_2^{k-1} \rightarrow_1^+$ is $(\eta(t), k-1)$ hence by *i.h.* we get $\tau \subseteq \rightarrow_1^+ \rightarrow_2^* \rightarrow_2^* \rightarrow_1^{h-1}$. For the suffix $\rightarrow_2^* \rightarrow_2^* \rightarrow_1^{h-1}$ it is the first component of the measure which decreases, since its starting term is obtained through a \rightarrow_1^+ -reduction from t , so one can apply the *i.h.* and get $\tau \subseteq \rightarrow_1^+ \rightarrow_1^+ \rightarrow_2^*$. ◀

The last abstract lemma we need is the one for the diagonal swaps, which has to take into account the phenomenon of a third reduction popping out in the right part, *i.e.* permutations of the form $\rightarrow_2 \rightarrow_1 \subseteq \rightarrow_1 \rightarrow_{2,3}^*$ (given for instance by Lemma 3.2.4). We prove a sort of generalization of Lemma 4.2 to this setting. This lemma is the key for the abstract factorization theorem of next section. It is an original contribution of the paper.

► **Lemma 4.4.** *If*

1. $\rightarrow_2 \rightarrow_1 \subseteq \rightarrow_1 \rightarrow_{2,3}^*$ and
 2. $\rightarrow_2 \rightarrow_3 \subseteq \rightarrow_3^+ \rightarrow_2^*$ and
 3. \rightarrow_3 is strongly normalizing
- then $\rightarrow_2^* \rightarrow_1^* \subseteq \rightarrow_{1,3}^* \rightarrow_2^*$.

Proof. First of all note that the second and third hypothesis allow to apply Lemma 4.3 and get $\rightarrow_{2,3}^* \subseteq \rightarrow_3^* \rightarrow_2^*$, let us call this fact property (#). The proof is by induction on (h, k) , where $\rightarrow_2^k \rightarrow_1^h$, ordered lexicographically. If $k = 0$ or $h = 0$ it is trivial. If $k > 0$ and $h > 0$ then applying the first hypothesis to the central part of $\tau : \rightarrow_2^{k-1} \rightarrow_2 \rightarrow_1 \rightarrow_1^{h-1}$ we get:

$$\tau \subseteq \rightarrow_2^{k-1} \rightarrow_1 \rightarrow_{2,3}^* \rightarrow_1^{h-1} \subseteq_{(\#)} \rightarrow_2^{k-1} \rightarrow_1 \rightarrow_3^* \rightarrow_2^* \rightarrow_1^{h-1}$$

Then by applying the *i.h.* to the prefix $\rightarrow_2^{k-1} \rightarrow_1$ we get:

$$\tau \subseteq \rightarrow_{1,3}^* \underbrace{\rightarrow_2^* \rightarrow_3^* \rightarrow_2^*}_{\rightarrow_{2,3}^*} \rightarrow_1^{h-1} \subseteq_{(\#)} \underbrace{\rightarrow_{1,3}^* \rightarrow_3^*}_{\rightarrow_{1,3}^*} \rightarrow_2^* \rightarrow_1^{h-1} \subseteq \rightarrow_{1,3}^* \rightarrow_2^* \rightarrow_1^{h-1}$$

Now by applying the *i.h.* to the suffix $\rightarrow_2^* \rightarrow_1^{h-1}$ we conclude with $\tau \subseteq \underbrace{\rightarrow_{1,3}^* \rightarrow_{1,3}^*}_{\rightarrow_{1,3}^*} \rightarrow_2^* \subseteq \rightarrow_{1,3}^* \rightarrow_2^*$ ◀

³ For the sake of simplicity the proof implicitly assumes that S is a finitely branching rewriting system. This assumption can be dropped by replacing the induction on $\eta(t)$ with an induction on the well-founded order on the elements of S induced by \rightarrow_1 (because of strong normalization). We learned this from Roberto Di Cosmo.

5 Abstract Standardization

In this section we prove that any square factorization system enjoys a factorization theorem. Let us recall the definition from Section 3.

► **Definition.** A **square factorization system** (SFS) S is given by a set $|S|$ and four reduction relations $(\rightsquigarrow_\bullet, \rightsquigarrow_\circ, \rightarrow_\bullet, \rightarrow_\circ)$ on $|S|$ s. t. the following conditions hold (remember that $\rightsquigarrow := \rightsquigarrow_\circ \cup \rightsquigarrow_\bullet$, $\rightarrow := \rightarrow_\circ \cup \rightarrow_\bullet$, $\rightarrow_\circ := \rightsquigarrow_\circ \cup \rightarrow_\bullet$, $\rightarrow_\bullet := \rightsquigarrow_\bullet \cup \rightarrow_\circ$ and $\rightarrow := \rightsquigarrow \cup \rightarrow$):

1. **Termination:** \rightsquigarrow_\circ and \rightarrow_\circ are strongly normalizing.
2. **Row-swap 1:** $\rightsquigarrow_\bullet \rightsquigarrow_\circ \subseteq \rightsquigarrow_\circ^+ \rightsquigarrow_\bullet^*$.
3. **Row-swap 2:** $\rightarrow_\bullet \rightarrow_\circ \subseteq \rightarrow_\circ^+ \rightarrow_\bullet^*$.
4. **Diagonal-swap 1:** $\rightarrow_\bullet \rightsquigarrow_\circ \subseteq \rightsquigarrow_\circ \rightarrow_\bullet^*$.
5. **Diagonal-swap 2:** $\rightsquigarrow_\bullet \rightarrow_\circ \subseteq \rightarrow_\circ \rightsquigarrow_\bullet^*$.

The proof relies on the abstract lemmas of the previous sections and is simple. We just lift the local permutations demanded to a SFS to sequences of steps (Points 1.a, 1.b, 2.a, 2.b below) and then glue them together (Points 1.c and 2.c).

► **Lemma 5.1** (Induced permutations of \bullet^* and \circ^*). *Let $S = \{\rightsquigarrow_\bullet, \rightsquigarrow_\circ, \rightarrow_\bullet, \rightarrow_\circ\}$ be a square factorization system. The following inclusions hold:*

- | | |
|---|---|
| 1) Swap of \rightarrow_\bullet^* and \rightarrow_\circ^*: | 2) Swap of $\rightsquigarrow_\bullet^*$ and \rightsquigarrow_\circ^*: |
| a) Diagonal: $\rightarrow_\bullet \rightsquigarrow_\circ \subseteq \rightarrow_\circ \rightarrow_\bullet^*$. | a) Diagonal: $\rightsquigarrow_\bullet \rightarrow_\circ \subseteq \rightarrow_\circ \rightsquigarrow_\bullet^*$. |
| b) Row: $\rightarrow_\bullet \rightarrow_\circ \subseteq \rightarrow_\circ \rightarrow_\bullet^*$. | b) Row: $\rightsquigarrow_\bullet \rightsquigarrow_\circ \subseteq \rightsquigarrow_\circ \rightsquigarrow_\bullet^*$. |
| c) Diagonal + Row: $\rightarrow_\bullet \rightarrow_\circ^* \subseteq \rightarrow_\circ^* \rightarrow_\bullet^*$. | c) Diagonal + Row: $\rightsquigarrow_\bullet \rightarrow_\circ^* \subseteq \rightarrow_\circ^* \rightsquigarrow_\bullet^*$. |

Proof. 1.a) Conditions 4 ($\rightarrow_\bullet \rightsquigarrow_\circ \subseteq \rightsquigarrow_\circ \rightarrow_\bullet^*$), 3 ($\rightarrow_\bullet \rightarrow_\circ \subseteq \rightarrow_\circ^+ \rightarrow_\bullet^*$) and 1 (\rightsquigarrow_\circ is strongly normalizing) for SFS satisfy the hypothesis of Lemma 4.4 (with $\rightarrow_1 = \rightsquigarrow_\circ$, $\rightarrow_2 = \rightarrow_\bullet$, $\rightarrow_3 = \rightarrow_\circ$, $\rightarrow_{2,3} = (\rightarrow_\bullet \cup \rightarrow_\circ) = \rightarrow$ and $\rightarrow_{1,3} = (\rightarrow_\circ \cup \rightsquigarrow_\circ) = \rightarrow_\circ$). Hence, we get that $\rightarrow_\bullet \rightsquigarrow_\circ^*$ is included in $\rightarrow_\circ^* \rightarrow_\bullet^*$.

1.b) Conditions 1 (\rightarrow_\circ is strongly normalizing) and 3 ($\rightarrow_\bullet \rightarrow_\circ \subseteq \rightarrow_\circ^+ \rightarrow_\bullet^*$) for SFS satisfy the hypothesis of Lemma 4.3 (with $\rightarrow_1 = \rightarrow_\circ$, $\rightarrow_2 = \rightarrow_\bullet$). Hence, we get $(\rightarrow_\circ \cup \rightarrow_\bullet)^* \subseteq \rightarrow_\circ^* \rightarrow_\bullet^*$. We conclude, since $\rightarrow_\bullet \rightsquigarrow_\circ^*$ is included in $(\rightarrow_\circ \cup \rightarrow_\bullet)^*$ and thus in $\rightarrow_\circ^* \rightarrow_\bullet^*$.

1.c) From 1.a and 1.b we get $\rightarrow_\bullet \rightsquigarrow_\circ \subseteq \rightarrow_\circ^* \rightarrow_\bullet^*$ and $\rightarrow_\bullet \rightarrow_\circ \subseteq \rightarrow_\circ^* \rightarrow_\bullet^*$, hence $\rightarrow_\bullet \rightarrow_\circ \subseteq \rightarrow_\circ^* \rightarrow_\bullet^*$. By Lemma 4.1 we get $(\rightarrow_\circ \cup \rightarrow_\bullet)^* \subseteq \rightarrow_\circ^* \rightarrow_\bullet^*$ and we conclude, since $\rightarrow_\bullet \rightarrow_\circ^* \subseteq (\rightarrow_\circ \cup \rightarrow_\bullet)^*$.

2.a) Conditions 5 ($\rightsquigarrow_\bullet \rightarrow_\circ \subseteq \rightarrow_\circ \rightsquigarrow_\bullet^*$), 2 ($\rightsquigarrow_\bullet \rightsquigarrow_\circ \subseteq \rightsquigarrow_\circ^+ \rightsquigarrow_\bullet^*$) and 1 (\rightsquigarrow_\circ is strongly normalizing) for SFS satisfy the hypothesis of Lemma 4.4 (with $\rightarrow_1 = \rightsquigarrow_\circ$, $\rightarrow_2 = \rightsquigarrow_\bullet$, $\rightarrow_3 = \rightsquigarrow_\circ$, $\rightarrow_{2,3} = (\rightsquigarrow_\bullet \cup \rightsquigarrow_\circ) = \rightsquigarrow$ and $\rightarrow_{1,3} = (\rightsquigarrow_\circ \cup \rightsquigarrow_\circ) = \rightsquigarrow_\circ$). Hence, we get that $\rightsquigarrow_\bullet \rightarrow_\circ^*$ is included in $\rightarrow_\circ^* \rightsquigarrow_\bullet^*$.

2.b) Conditions 1 (\rightsquigarrow_\circ is strongly normalizing) and 2 ($\rightsquigarrow_\bullet \rightsquigarrow_\circ \subseteq \rightsquigarrow_\circ^+ \rightsquigarrow_\bullet^*$) for SFS satisfy the hypothesis of Lemma 4.3 (with $\rightarrow_1 = \rightsquigarrow_\circ$, $\rightarrow_2 = \rightsquigarrow_\bullet$). Hence, we get $(\rightsquigarrow_\circ \cup \rightsquigarrow_\bullet)^* \subseteq \rightsquigarrow_\circ^* \rightsquigarrow_\bullet^*$. We conclude, since $\rightsquigarrow_\bullet \rightarrow_\circ^*$ is included in $(\rightsquigarrow_\circ \cup \rightsquigarrow_\bullet)^*$ and thus in $\rightsquigarrow_\circ^* \rightsquigarrow_\bullet^*$.

2.c) From 2.a and 2.b we get $\rightsquigarrow_\bullet \rightarrow_\circ \subseteq \rightarrow_\circ^* \rightsquigarrow_\bullet^*$ and $\rightsquigarrow_\bullet \rightsquigarrow_\circ \subseteq \rightsquigarrow_\circ^* \rightsquigarrow_\bullet^*$, hence $\rightsquigarrow_\bullet \rightarrow_\circ \subseteq \rightarrow_\circ^* \rightsquigarrow_\bullet^*$. By Lemma 4.1 we get $(\rightsquigarrow_\circ \cup \rightsquigarrow_\bullet)^* \subseteq \rightarrow_\circ^* \rightsquigarrow_\bullet^*$ and we conclude, since $\rightsquigarrow_\bullet \rightarrow_\circ^* \subseteq (\rightsquigarrow_\circ \cup \rightsquigarrow_\bullet)^*$. ◀

It is now easy to conclude.

► **Theorem 5.2.** *Let $S = \{\rightsquigarrow_\bullet, \rightsquigarrow_\circ, \rightarrow_\bullet, \rightarrow_\circ\}$ be a square factorization system. Then:*

1. **Swap of \bullet and \circ^* :** $\rightarrow_\bullet \rightarrow_\circ^* \subseteq \rightarrow_\circ^* \rightarrow_\bullet^*$.
2. **Factorization:** $\rightarrow^* \subseteq \rightarrow_\circ^* \rightarrow_\bullet^*$.

Proof. 1. From points 1.c and 2.c of Lemma 5.1 we get $\rightarrow_\bullet \rightarrow_\circ^* \subseteq \rightarrow_\circ^* \rightarrow_\bullet^*$ and $\rightsquigarrow_\bullet \rightarrow_\circ^* \subseteq \rightarrow_\circ^* \rightsquigarrow_\bullet^*$, hence $\rightarrow_\bullet \rightarrow_\circ^* \subseteq \rightarrow_\circ^* \rightarrow_\bullet^*$.

2. By induction on the length k of \rightarrow^* . The case $k = 0$ is trivial, then let $k > 0$. By *i.h.* we get $\rightarrow \rightarrow^{k-1} \subseteq \rightarrow \rightarrow_{\circ}^* \rightarrow_{\bullet}^*$. If the first step is a \rightarrow_{\circ} step then there is nothing to prove. Otherwise, by the previous point we get $\rightarrow_{\bullet} \rightarrow_{\circ}^* \rightarrow_{\bullet}^* \subseteq \rightarrow_{\circ}^* \rightarrow_{\bullet}^* \rightarrow_{\bullet}^*$ and conclude. \blacktriangleleft

All SFSs in the paper are composed of four rules. But some substitution calculi have many rules, in particular a set of rules for substitutions. To apply the theorem is enough to identify this set with \rightarrow , and to split each rule accordingly. Similarly, there can also be various rules for creating substitutions, whose union would be \rightsquigarrow . The axioms for a SFS are symmetric, so the choice between \rightarrow or \rightsquigarrow for one or the other set does not really matter.

6 The value-substitution calculus λ_{vsub}

The value-substitution calculus λ_{vsub} has the same syntax as λ_{sub} , but we distinguish the syntactic category of values:

$$v = x \mid \lambda x.t$$

The rewriting relation $\rightarrow_{\lambda_{\text{vsub}}}$ is defined as the union of \rightarrow_{dB} (defined as before) and \rightarrow_{vs} , which is the context closure of:

$$t[x/vL] \mapsto_{\text{vs}} t\{x/v\}L$$

Remark rule \mapsto_{vs} : there is no typo, in the left-hand side L is inside $[x/vL]$ and in the right-hand side it is outside $\{x/v\}$. A detailed account of the value-substitution calculus, which is confluent, can be found in [8]. For λ_{vsub} we are going to prove two different factorization theorems, both based on the following property.

► **Lemma 6.1** ([8]). \rightarrow_{dB} and \rightarrow_{vs} are strongly normalizing and confluent.

The first factorization scheme for $\rightarrow_{\lambda_{\text{vsub}}}$ is given by weak reduction (*i.e.* reductions out of lambdas) and non-weak reductions, playing for λ_{vsub} the role of head and internal reductions for λ_{sub} . **Weak contexts** are contexts whose hole is not under an abstraction:

$$W ::= [\cdot] \mid W \ t \mid t \ W \mid W[x/t] \mid t[x/W]$$

Weak reduction \rightarrow_{\circ} is defined as the union of $\rightarrow_{\text{dB}\circ}$ and $\rightarrow_{\text{vs}\circ}$, which are obtained as the closure by *weak* contexts of \mapsto_{dB} and \mapsto_{vs} . Note that the reductions $\rightarrow_{\text{vs}\circ}$ and $\rightarrow_{\text{dB}\circ}$ are strongly normalizing because their unrestricted versions are. Weak reduction \rightarrow_{\circ} has the diamond property [8].

In order to obtain a square factorization system we need to define the complements of \rightarrow_{\circ} , $\rightarrow_{\text{dB}\circ}$ and $\rightarrow_{\text{vs}\circ}$. We define \rightarrow_{\bullet} as the union of $\rightarrow_{\text{dB}\bullet} := \rightarrow_{\text{dB}} \setminus \rightarrow_{\text{dB}\circ}$ and $\rightarrow_{\text{vs}\bullet} := \rightarrow_{\text{vs}} \setminus \rightarrow_{\text{vs}\circ}$, which are easily seen to be also definable via **non-weak contexts**, given by $\overline{W} ::= \lambda x.C \mid W[\overline{W}]$. Two examples:

$$(\lambda x.t)[x/v[y/u]] \ s \rightarrow_{\text{vs}\circ} (\lambda x.t)\{x/v\}[y/u] \ s \qquad t \ \lambda x.(s[x/v]) \rightarrow_{\text{vs}\bullet} t \ \lambda x.(s\{x/v\})$$

The following theorem is proved by means of a substitution lemma and an induction on the rewriting relations (details in [2]).

► **Theorem 6.2** (weak factorization). *The following inclusions hold:*

1. **Row 1:** $\rightarrow_{\text{dB}\bullet} \rightarrow_{\text{dB}\circ} \subseteq \rightarrow_{\text{dB}\circ} \rightarrow_{\text{dB}\bullet} \cup \rightarrow_{\text{dB}\circ} \rightarrow_{\text{dB}\circ}$.
2. **Row 2:** $\rightarrow_{\text{vs}\bullet} \rightarrow_{\text{vs}\circ} \subseteq \rightarrow_{\text{vs}\circ} \rightarrow_{\text{vs}\bullet}^*$.

3. **Diagonal 1:** $\rightarrow_{\text{vs}\bullet} \rightarrow_{\text{dB}\circ} \subseteq \rightarrow_{\text{dB}\circ} \rightarrow_{\text{vs}\circ} \cup \rightarrow_{\text{dB}\circ} \rightarrow_{\text{vs}\bullet}$.

4. **Diagonal 2:** $\rightarrow_{\text{dB}\bullet} \rightarrow_{\text{vs}\circ} \subseteq \rightarrow_{\text{vs}\circ} \rightarrow_{\text{dB}\bullet}^*$.

Thus $(\rightarrow_{\text{dB}\bullet}, \rightarrow_{\text{dB}\circ}, \rightarrow_{\text{vs}\bullet}, \rightarrow_{\text{vs}\circ})$ is a SFS and $\rightarrow_{\lambda_{\text{vsub}}}^* = \rightarrow_{\circ}^* \rightarrow_{\bullet}^*$.

We invite the reader to compare the diagrams of λ_{sub} (Lemma 3.2) and λ_{vsub} (Theorem 6.2): they are similar and yet different, despite both are SFS.

At this point we could iterate the factorization as for λ_{sub} , but we prefer to show another factorization scheme obtained by combining head reduction and weak reduction in a new interesting way. The following grammar defines stratified-weak contexts SW :

$$SW ::= W \mid SW \ t \mid \lambda x. SW \mid SW[t/x]$$

Note that equivalently $SW ::= H[W]$. Stratified-weak reduction \rightarrow_{\circ} is defined as the union of $\rightarrow_{\text{dB}\circ}$ and $\rightarrow_{\text{vs}\circ}$ the closures by stratified-weak contexts \mapsto_{dB} and \mapsto_{vs} . Stratified-weak reduction \rightarrow_{\circ} has the diamond property [8]. We also need the complements of $\rightarrow_{\circ}, \rightarrow_{\text{dB}\circ}$ and $\rightarrow_{\text{vs}\circ}$. Let us set: $\neg_{\bullet} := \rightarrow \setminus \rightarrow_{\circ}$, $\neg_{\text{dB}\bullet} := \rightarrow_{\text{dB}} \setminus \rightarrow_{\text{dB}\circ}$ and $\neg_{\text{vs}\bullet} := \rightarrow_{\text{vs}} \setminus \rightarrow_{\text{vs}\circ}$. Two examples:

$$\lambda x.(s[x/v]) \rightarrow_{\text{vs}\circ} \lambda x.(s\{x/v\}) \quad t[y/\lambda x.(s[x/v])] \rightarrow_{\text{vs}\bullet} t[y/\lambda x.(s\{x/v\})]$$

The following theorem is proved by means of a substitution lemma and by induction on the rewriting relations (details in [2]).

► **Theorem 6.3** (stratified-weak factorization). *The following inclusions hold:*

1. **Row 1:** $\neg_{\text{dB}\bullet} \neg_{\text{dB}\circ} \subseteq \neg_{\text{dB}\circ} \neg_{\text{dB}\bullet} \cup \neg_{\text{dB}\circ} \neg_{\text{dB}\circ}$.

2. **Row 2:** $\neg_{\text{vs}\bullet} \neg_{\text{vs}\circ} \subseteq \neg_{\text{vs}\circ} \neg_{\text{vs}\bullet}^* \cup \neg_{\text{vs}\circ} \neg_{\text{vs}\circ} \neg_{\text{vs}\bullet}^*$.

3. **Diagonal 1:** $\neg_{\text{vs}\bullet} \neg_{\text{dB}\circ} \subseteq \neg_{\text{dB}\circ} \neg_{\text{vs}\bullet} \cup \neg_{\text{dB}\circ} \neg_{\text{vs}\circ}$.

4. **Diagonal 2:** $\neg_{\text{dB}\bullet} \neg_{\text{vs}\circ} \subseteq \neg_{\text{vs}\circ} \neg_{\text{dB}\bullet}^* \cup \neg_{\text{vs}\circ} \neg_{\text{dB}\bullet}^*$.

Thus $(\neg_{\text{dB}\bullet}, \neg_{\text{dB}\circ}, \neg_{\text{vs}\bullet}, \neg_{\text{vs}\circ})$ is a SFS and $\rightarrow_{\lambda_{\text{vsub}}}^* = \neg_{\circ}^* \rightarrow_{\bullet}^*$.

Note that once again the permutations are different from the previous cases, and yet caught by the axioms for a SFS. The two factorization theorems for λ_{vsub} have been used in [8] to give a new operational characterization of call-by-value solvability.

7 The linear substitution calculus λ_{lsub}

In this section we apply the factorization theorem to a finer calculus of explicit substitutions, the linear substitution calculus λ_{lsub} , a slight variation over a calculus introduced by Robin Milner in [29], which can also be seen as a simplification of the structural λ -calculus λ_j [5]. The linear substitution calculus has the same syntax as λ_{sub} . The rewriting rules of λ_{lsub} are \rightarrow_{dB} (defined as before), \rightarrow_{ls} and \rightarrow_{gc} , which are defined as the context closures of:

$$C[x][x/u] \mapsto_{\text{ls}} C[u][x/u] \quad t[x/u] \mapsto_{\text{gc}} t \quad \text{if } x \notin \text{fv}(t)$$

Rule \mapsto_{ls} uses contexts: whenever we write $C[x][x/u]$ we implicitly assume that C does not capture x , despite the fact that in general (for instance in the context closure of the rules) contexts may capture variables. We also note $\rightarrow_{\lambda_{\text{lsub}}} = \rightarrow_{\text{dB}} \cup \rightarrow_{\text{ls}} \cup \rightarrow_{\text{gc}}$. Note that λ_{lsub} is obtained from λ_{sub} by decomposing the substitution rule in two more atomic rules. The difference between λ_{lsub} and the original calculus of Milner [29] is rule \rightarrow_{dB} , which is not at a distance in Milner's calculus (*i.e.* it has the form of the B-rule in the introduction).

The linear substitution calculus enjoys all properties demanded to explicit substitution calculi, obtained by easy adaptations of the proofs for λ_j in [5]. In particular, it is confluent and preserves β -strong normalization. We will use the following property:

► **Lemma 7.1.** $\rightarrow_{\text{dB}}, \rightarrow_{\text{ls}}, \rightarrow_{\text{gc}}$ are strongly normalizing and confluent.

Another important property is that garbage collection can be postponed.

► **Lemma 7.2.** Let $\rightarrow_{\neg\text{gc}}$ be $\rightarrow_{\text{dB}} \cup \rightarrow_{\text{ls}}$, i.e. the complement of \rightarrow_{gc} with respect to $\rightarrow_{\lambda_{\text{sub}}}$.

1. **Local postponement of garbage collection:** $\rightarrow_{\text{gc}} \rightarrow_{\neg\text{gc}} \subseteq \rightarrow_{\neg\text{gc}} \rightarrow_{\text{gc}}^*$.
2. **Postponement of garbage collection:** $\rightarrow_{\lambda_{\text{sub}}}^* \subseteq \rightarrow_{\neg\text{gc}}^* \rightarrow_{\text{gc}}^*$.

Proof. 1) By induction on \rightarrow_{gc} . 2) It follows from the previous point and Lemma 4.2. ◀

Interestingly, if one defines head reduction as for λ_{sub} , i.e. as the closure by head contexts of the root rewriting rules, the factorization theorem does not hold. Consider this reduction:

$$x[x/y[y/z]][z/u] \rightarrow_{\text{ls}} x[x/z[y/z]][z/u] \rightarrow_{\text{ls}} x[x/u[y/z]][z/u]$$

the first step has box level 1, while the second has box level 0, but they cannot be permuted⁴. As we will see later, the notion of level of the linear substitution calculus is more subtle.

In λ_{sub} it is possible to reformulate what in the Linear Logic literature is called *linear head reduction* [23], and to prove factorization of $\rightarrow_{\neg\text{gc}}$ with respect to it. Note that in a \rightarrow_{ls} step contexts are involved twice: to close the rule and to select the variable occurrence to substitute for. For a head *linear* \rightarrow_{ls} step both contexts have to be head.

► **Definition 7.3** (head linear reduction). **Head linear reduction** \rightarrow_{\circ} is defined as the union of $\rightarrow_{\text{dB}\circ} := H[\mapsto_{\text{dB}}]$ and $\rightarrow_{\text{ls}\circ} := H[\hat{\rightarrow}_{\text{ls}}]$, where $\hat{\rightarrow}_{\text{ls}}$ is given by:

$$H[x][x/u] \hat{\rightarrow}_{\text{ls}} H[u][x/u]$$

Internal linear reduction \rightarrow_{\bullet} is defined as the union of $\rightarrow_{\text{dB}\bullet} := \rightarrow_{\text{dB}} \setminus \rightarrow_{\text{dB}\circ}$ and $\rightarrow_{\text{ls}\bullet} := \rightarrow_{\text{ls}} \setminus \rightarrow_{\text{ls}\circ}$. Two examples:

$$(y\ y)[y/z] \rightarrow_{\text{ls}\circ} (z\ y)[y/z] \qquad (y\ y)[y/z] \rightarrow_{\text{ls}\bullet} (y\ z)[y/z]$$

With these definitions we get the following theorem, proved at the end of the section:

► **Theorem 7.4.** $(\rightarrow_{\text{dB}\bullet}, \rightarrow_{\text{dB}\circ}, \rightarrow_{\text{ls}\bullet}, \rightarrow_{\text{ls}\circ})$ is a SFS for $\rightarrow_{\neg\text{gc}}$, and so $\rightarrow_{\neg\text{gc}}^* \subseteq \rightarrow_{\circ}^* \rightarrow_{\bullet}^*$.

To our knowledge this is the first result formally proving factorization with respect to linear head reduction. We recall that head linear reduction can be seen as an abstraction of Krivine's abstract machine [14]. Therefore, when β -reduction is decomposed in more atomic rules, the factorization theorem becomes a method to derive abstract machines.

The relation between head linear reduction and the head reduction of λ -calculus has been studied in depth in [7].

It is possible to generalize head linear reduction \rightarrow_{\circ} into a family $\{\rightarrow_{\circ}^i\}_{i \in \mathbb{N}}$ of reductions by levels, in which \rightarrow_{\circ} corresponds to \rightarrow_{\circ}^0 . We still use contexts to define the level of a step, but the notion of level used here differs from the one used for λ_{sub} : the index counts the arguments of applications only. Contexts whose hole is inside a substitution are noted H_{∞} . **Leveled contexts** are defined as follows:

$$H_0 ::= H \qquad H_{i+1} ::= t\ H_i \mid H_0[H_{i+1}] \qquad H_{\infty} ::= t\ [x/C] \mid H_i[H_{\infty}]$$

⁴ A note for Proof-Nets experts: this phenomenon is at first sight surprising, but it is not due to our formalism and we are not the firsts to point it out: working at the same time with small-steps rules (as in λ_{sub}) and freely permutating contraction with !-boxes (which is implicit using terms) breaks standardization by levels also in a traditional Proof-Nets syntax, see Tranquilli's thesis [33], p. 133.

In order to define linear reduction by levels we fix some conventions. We will need to sum indexes, and for ∞ we use the equalities $\infty + i = \infty$ and $\infty + \infty = \infty$. Moreover, we note $\mathbb{N}^\infty = \mathbb{N} \cup \{\infty\}$ and consider $i < \infty$ for all $i \in \mathbb{N}$.

► **Definition 7.5.** The **linear reduction** $\overset{i}{\rightarrow}$ is the union of $\overset{i}{\rightarrow}_{\text{dB}}$ and $\overset{i}{\rightarrow}_{1\text{s}}$, defined as follows. For $i \in \mathbb{N}^\infty$ the reduction $\overset{i}{\rightarrow}_{\text{dB}}$ is the closure by H_i -contexts of \mapsto_{dB} . The case of $\rightarrow_{1\text{s}}$ is a bit more complex. Let $\overset{j}{\mapsto}_{1\text{s}}$ be defined by the rule:

$$H_j[x][x/u] \overset{j}{\mapsto}_{1\text{s}} H_j[u][x/u]$$

Then define $\overset{i}{\rightarrow}_{1\text{s}} := H_l[\overset{j}{\mapsto}_{1\text{s}}]$, where $i = l + j$. Define also $\overset{>i}{\rightarrow}_{1\text{s}} := \bigcup_{j>i} \overset{j}{\rightarrow}_{1\text{s}}$ and similarly for $\overset{>i}{\rightarrow}_{\text{dB}}$, $\overset{>i}{\rightarrow}$ and for \geq . Some examples of $\overset{i}{\rightarrow}_{1\text{s}}$ reductions:

$$\begin{aligned} (y \ y)[y/z] \overset{0}{\rightarrow}_{1\text{s}} (z \ y)[y/z] & \quad (y \ y)[y/z] \overset{1}{\rightarrow}_{1\text{s}} (y \ z)[y/z] & \quad y[y/z][z/s] \overset{\infty}{\rightarrow}_{1\text{s}} y[y/s][z/s] \\ (x \ (y \ y)[y/z]) \overset{2}{\rightarrow}_{1\text{s}} (x \ (y \ z)[y/z]) & \quad (x \ (y \ y)[y/z])[z/s] \overset{\infty}{\rightarrow}_{1\text{s}} (x \ (y \ y)[y/s])[z/s] \end{aligned}$$

Note that when $t \overset{i}{\rightarrow}_{1\text{s}} s$ then exists H_i s.t. $t = H_i[x] \overset{i}{\rightarrow}_{1\text{s}} H_i[u]$ with respect to some substitution $[x/u]$ in H_i .

The following lemma gives a detailed analysis of leveled diagrams for the linear substitution calculus. It is proved by an analysis of diagrams relying on some technical lemmas on leveled contexts (details in [2]).

► **Lemma 7.6.** *Let $i < j < \infty$. Then:*

- 1) **Row 1a:** $\overset{j}{\rightarrow}_{\text{dB}} \overset{i}{\rightarrow}_{\text{dB}} \subseteq \overset{j}{\rightarrow}_{\text{dB}} \overset{i}{\rightarrow}_{\text{dB}} \cup \overset{i}{\rightarrow}_{\text{dB}} \overset{\infty}{\rightarrow}_{\text{dB}}$.
- 2) **Row 1b:** $\overset{\infty}{\rightarrow}_{\text{dB}} \overset{i}{\rightarrow}_{\text{dB}} \subseteq \overset{\infty}{\rightarrow}_{\text{dB}} \overset{\infty}{\rightarrow}_{\text{dB}}$.
- 3) **Row 2a:** $\overset{j}{\rightarrow}_{1\text{s}} \overset{i}{\rightarrow}_{1\text{s}} \subseteq \overset{i}{\rightarrow}_{1\text{s}} \overset{j}{\rightarrow}_{1\text{s}}$.
- 4) **Diagonal 1a:** $\overset{j}{\rightarrow}_{1\text{s}} \overset{i}{\rightarrow}_{\text{dB}} \subseteq \overset{i}{\rightarrow}_{\text{dB}} \overset{j}{\rightarrow}_{1\text{s}} \cup \overset{i}{\rightarrow}_{\text{dB}} \overset{\infty}{\rightarrow}_{1\text{s}}$.
- 5) **Diagonal 1b:** $\overset{\infty}{\rightarrow}_{1\text{s}} \overset{i}{\rightarrow}_{\text{dB}} \subseteq \overset{i}{\rightarrow}_{\text{dB}} \overset{\infty}{\rightarrow}_{1\text{s}}$.
- 6) **Diagonal 2a:** $\overset{j}{\rightarrow}_{\text{dB}} \overset{i}{\rightarrow}_{1\text{s}} \subseteq \overset{i}{\rightarrow}_{1\text{s}} \overset{j}{\rightarrow}_{\text{dB}}$.
- 7) **Row 2b:** $\overset{\infty}{\rightarrow}_{1\text{s}} \overset{i}{\rightarrow}_{1\text{s}} \subseteq \overset{i}{\rightarrow}_{1\text{s}} \overset{\infty}{\rightarrow}_{1\text{s}} \cup \overset{i}{\rightarrow}_{1\text{s}} \overset{i+k}{\rightarrow}_{1\text{s}} \overset{\infty}{\rightarrow}_{1\text{s}} \cup \overset{i}{\rightarrow}_{1\text{s}} \overset{\infty}{\rightarrow}_{1\text{s}} \overset{\infty}{\rightarrow}_{1\text{s}}$, with $k \in \mathbb{N}$.
- 8) **Diagonal 2b:** $\overset{\infty}{\rightarrow}_{\text{dB}} \overset{i}{\rightarrow}_{1\text{s}} \subseteq \overset{i}{\rightarrow}_{1\text{s}} \overset{\infty}{\rightarrow}_{\text{dB}} \cup \overset{i}{\rightarrow}_{1\text{s}} \overset{i+k}{\rightarrow}_{\text{dB}} \overset{\infty}{\rightarrow}_{\text{dB}} \cup \overset{i}{\rightarrow}_{1\text{s}} \overset{\infty}{\rightarrow}_{\text{dB}} \overset{\infty}{\rightarrow}_{\text{dB}}$, with $k \in \mathbb{N}$.

The lemma implies factorization by levels analogous to the one of λ_{sub} (but with respect to different leveled reductions).

► **Corollary 7.7** (Factorization by levels). $(\overset{>i}{\rightarrow}_{\text{dB}}, \overset{i}{\rightarrow}_{\text{dB}}, \overset{>i}{\rightarrow}_{1\text{s}}, \overset{i}{\rightarrow}_{1\text{s}})$ is a SFS for every $i \in \mathbb{N}$, and so $\overset{\geq i}{\rightarrow} * \subseteq \overset{i}{\rightarrow} * \overset{>i}{\rightarrow} *$.

Proof. The first axiom (termination) follows from Lemma 7.1. The second from points 1 and 2 of Lemma 7.6. The third from points 3 and 7. The fourth from points 4 and 5. The fifth from points 6 and 8. ◀

We can now easily prove Theorem 7.4, just note that $(\overset{\bullet}{\rightarrow}_{\text{dB}}, \overset{\bullet}{\rightarrow}_{\text{dB}}, \overset{\bullet}{\rightarrow}_{1\text{s}}, \overset{\bullet}{\rightarrow}_{1\text{s}})$ is nothing but $(\overset{>0}{\rightarrow}_{\text{dB}}, \overset{0}{\rightarrow}_{\text{dB}}, \overset{>0}{\rightarrow}_{1\text{s}}, \overset{0}{\rightarrow}_{1\text{s}})$. We conclude with a standardization theorem. Define the **linear level** of a term t as the maximum $i \in \mathbb{N}$ s.t. $t = H_i[u]$ for some u .

► **Theorem 7.8** (Standardization by levels). *Let $t \rightarrow_{\lambda_{1\text{sub}}}^* u$ and k is the linear level of u . Then $t \overset{0}{\rightarrow} * \overset{1}{\rightarrow} * \dots \overset{k-1}{\rightarrow} * \overset{k}{\rightarrow} * \overset{\infty}{\rightarrow} * \rightarrow_{\text{gc}}^* u$.*

Proof. Remark that $\xrightarrow{\infty}$ and \rightarrow_{gc} cannot change the linear level of a term and that $t \xrightarrow{i} v$ for i finite implies that both t and v have linear level at least i . From Lem. 7.2.2 we get that there exists v s.t. $t \xrightarrow{*}_{\text{gc}} v \xrightarrow{*}_{\text{gc}} u$. By the remark v has linear level k , the same of u . By iterating the use of Corollary 7.7 as in the proof of Corollary 3.7 we factor $\xrightarrow{*}_{\text{gc}}$ as $t \xrightarrow{0} * \xrightarrow{1} * \dots \xrightarrow{k-1} * \xrightarrow{k} * \xrightarrow{>k} * v$. Now, consider $\xrightarrow{>k} *$. By the remark $\xrightarrow{>k} *$ has to be a $\xrightarrow{\infty} *$ sequence and so we conclude. \blacktriangleleft

Theorem 7.8 provides a sharp decomposition of λ_{sub} -reductions. It is easy to prove that \xrightarrow{j} has the diamond property, for every $j \in \mathbb{N}$. However, the theorem can certainly be strengthened, as it says nothing about $\xrightarrow{\infty}$ (which does not enjoy the diamond property), whose analysis requires further factorizations, left to future work.

8 Conclusions

We presented an abstract factorization theorem for explicit substitution (ES) calculi, with applications to ES-calculi related to Proof-Nets. The abstract technique is simple. The theorem is applied more than once to every calculus, getting different factorizations, showing the flexibility of the abstract approach. It captures both call-by-name and call-by-value dynamics. We are convinced that the theorem applies to many other calculi, not related to Proof-Nets, despite this verification is left to future work.

We believe that our study of linear head reduction is a contribution: our presentation is simpler than the usual one [14], and the lack of a rewriting characterization for this important notion was a hole in the literature. A more detailed study of the linear substitution calculus, by means of residuals and permutation equivalence is an interesting research direction. It is also interesting to find other rewriting properties which can take advantage of the local termination nature of explicit substitution calculi.

Acknowledgements I want to thank Delia Kesner for many discussions on the subject and for her encouragement to try to publish this work. I also want to thank Flavio C. de Moura, Luca Paolini, Paul-André Melliès, Willem Heijltjes, and the anonymous referees.

References

- 1 M. Abadi, L. Cardelli, P. L. Curien, and J. J. Levy. Explicit substitutions. *Journal of Functional Programming*, 1:31–46, 1991.
- 2 B. Accattoli. An abstract factorisation theorem for explicit substitutions (extended version). available at <https://sites.google.com/site/beniaminoaccattoli/tech-report.pdf>.
- 3 B. Accattoli. *Jumping around the box: graphical and operational studies on Lambda Calculus and Linear Logic*. Ph.D. Thesis, Università di Roma La Sapienza, 2011.
- 4 B. Accattoli and S. Guerrini. Jumping boxes. representing λ -calculus boxes by jumps. In *CSL*, volume 5771 of *LNCS*, pages 55–70. Springer, 2009.
- 5 B. Accattoli and D. Kesner. The structural λ -calculus. In *Computer Science Logic (CSL)*, volume 6247 of *Lecture Notes in Computer Science*, pages 381–395. Springer, 2010.
- 6 B. Accattoli and D. Kesner. The permutative λ -calculus. In *LPAR*, pages 23–36, 2012.
- 7 B. Accattoli and U. Dal Lago. On the invariance of the unitary cost model for head reduction. Accepted at RTA 2012.
- 8 B. Accattoli and L. Paolini. Call-by-value solvability, revisited. Accepted to FLOPS 2012, available at <https://sites.google.com/site/beniaminoaccattoli/solvability.pdf>.

- 9 H. Barendregt. *The Lambda Calculus: Its Syntax and Semantics*. North-Holland, Amsterdam, revised edition, 1984. (First edition, 1981).
- 10 R. Di Cosmo and A. Piperno. Expanding extensional polymorphism. In *TLCA*, pages 139–153, 1995.
- 11 Pierre-Louis Curien, Thérèse Hardin, and Jean-Jacques Lévy. Confluence properties of weak and strong calculi of explicit substitutions. *J. ACM*, 43(2):362–397, 1996.
- 12 V. Danos, H. Herbelin, and L. Regnier. Game semantics & abstract machines. In *LICS*, pages 394–405, 1996.
- 13 V. Danos and L. Regnier. Reversible, irreversible and optimal lambda-machines. *Electr. Notes Theor. Comput. Sci.*, 3, 1996.
- 14 V. Danos and L. Regnier. Head linear reduction. Technical report, 2004.
- 15 D. de Carvalho, M. Pagani, and L. Tortora de Falco. A semantic measure of the execution time in linear logic. *TCS, Special issue Girard's Festschrift*, 412(20):1884–1902, 2011.
- 16 T. Ehrhard and L. Regnier. Böhm trees, krivine's machine and the taylor expansion of lambda-terms. In *CiE*, pages 186–197, 2006.
- 17 A. Geser. *Relative Termination*. Ph.d. thesis, Fakultät für Mathematik und Informatik, Universität Passau, Germany, 1990.
- 18 Georges Gonthier, Jean-Jacques Lévy, and Paul-André Melliès. An abstract standardisation theorem. In *LICS*, pages 72–81, 1992.
- 19 H. Herbelin and S. Zimmermann. An operational account of call-by-value minimal and classical lambda-calculus in "natural deduction" form. In *TLCA*, pages 142–156, 2009.
- 20 R. E. Møgelberg J. Egger and A. Simpson. Enriching an effect calculus with linear types. In *CSL*, pages 240–254, 2009.
- 21 D. N. Turner J. Maraist, M. Odersky and P. Wadler. Call-by-name, call-by-value, call-by-need and the linear lambda calculus. *Theor. Comput. Sci.*, 228(1-2):175–210, 1999.
- 22 D. Kesner. The theory of calculi with explicit substitutions revisited. In *CSL*, pages 238–252, 2007.
- 23 G. Mascari and M. Pedicini. Head linear reduction and pure proof net extraction. *Theor. Comput. Sci.*, 135(1):111–137, 1994.
- 24 D. Mazza. Pi et lambda. Une étude sur la traduction des lambda-termes dans le pi-calcul. Memoire de DEA (in french), 2003.
- 25 P.-A. Melliès. Typed lambda-calculi with explicit substitutions may not terminate. In *TLCA*, pages 328–334, 1995.
- 26 P.-A. Melliès. *Description axiomatique des systèmes de réécriture*. Phd thesis, Université Paris VII, 1996.
- 27 P.-A. Melliès. A factorisation theorem in rewriting theory. In *Category Theory and Computer Science*, pages 49–68, 1997.
- 28 P.-A. Melliès. Axiomatic rewriting theory I: A diagrammatic standardization theorem. In *Processes, Terms and Cycles*, pages 554–638, 2005.
- 29 R. Milner. Local bigraphs and confluence: Two conjectures: (extended abstract). *Electr. Notes Theor. Comput. Sci.*, 175(3):65–73, 2007.
- 30 M. Takahashi. Parallel reductions in lambda-calculus. *Inf. Comput.*, 118(1):120–127, 1995.
- 31 Terese. *Term Rewriting Systems*, volume 55 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2003.
- 32 K. Terui. Light affine lambda calculus and polynomial time strong normalization. *Arch. Math. Log.*, 46(3-4):253–280, 2007.
- 33 P. Tranquilli. *Nets Between Determinism and Nondeterminism*. Ph.d. thesis, Università degli Studi Roma Tre/Université Paris Diderot (Paris 7), 2009.